CLAIMS

1. A method for improving execution performance of a repeated sequence of instructions that provide a function and having external access points that are external entry and external exit points, comprising the steps of:

determining at least one instruction, from the sequence of instructions, that is necessary to be executed for less than all repetitions of the sequence of instructions; and

modifying the sequence of instructions to isolate the one instruction from only some of the repetitions of the sequence of instructions.

2. The method of claim 1, wherein:

said modifying includes the step of inserting at least one internal access point within the sequence of instructions and thereby partitioning the sequence of instructions into multiple segments, and having one of the

multiple segments including the one instruction and executing for fewer times than the number of executions of another of the multiple segments.

3. The method of claim 2, wherein

said inserting step inserts the one internal access point as an internal recursive entry point.

4. The method of claim 3, wherein

said modifying includes the step of moving the one instruction from outside of the one of the multiple segments to within the one of the multiple segments and between one of the external access points and the internal recursive access point.

5. The method of claim 2, wherein

said modifying includes the step of moving the one instruction from

outside of the one of the multiple segments to within the one of the multiple segments and between one of the external access points and the internal access point.

6. The method of claim 1, wherein

said modifying includes the step of rescheduling the one instruction closer in sequence of execution to one of the external access points.

7. A computer readable storage media having computer readable code physically implementing a method of improving execution performance of a sequence of instructions, the code including statements for performing the method of claim 1.

8. A computer readable storage media having computer readable code physically implementing a method of improving execution

performance of a sequence of instructions, the code including statements for performing the method of claim 2.

9. A computer readable storage media having computer readable code physically implementing a method of improving execution performance of a recursive sequence of instructions, the code including statements for performing the method of claim 5.

10. A computer readable storage media having computer readable code physically implementing a method of improving execution performance of a sequence of instructions, the code including statements for performing the method of claim 6.

11. A computer system including the computer readable storage media of claim 7, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions of the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

12. A computer system including the computer readable storage media of claim 8, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions of the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

13. A computer system including the computer readable storage media of claim 9, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions of the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

14. A computer system including the computer readable storage media of claim 10, further comprising:

at least one processing unit coupled to said computer readable storage media for executing the sequence of instructions and the computer readable code; and

said computer readable storage media including at least one of volatile and non-volatile memory.

15. A method of machine executing a called program of a repeated sequence of instructions having at least one instruction that is necessary

to be executed for less than all repetitions of the program, comprising:

executing at least some of the sequence of instructions from an externally called entry point in the program initially;

thereafter repeatedly calling the program;

in response to said repeatedly calling, executing only some of the sequence of instructions;

thereafter exiting the program from an exit point; and

controlling at least one of said steps of executing with an internal access point other than the entry point and the exit point to isolate the one instruction within the sequence of instructions from at least one of said repeatedly calling and to execute the one instruction a number of times fewer than the total number of executions of the entire sequence of instructions.

16. A method of machine executing according to claim 15, wherein:

said first-mentioned executing, includes executing the one

instruction;

said internal access point is an internal recursive entry point scheduled after the one instruction in the sequence of instructions; and

said second-mentioned executing recursively starts from the internal recursive entry point.

17. A method of processing, comprising:

providing a sequence of instructions repeatable to perform a function and having at least one instruction that is necessary to be executed for less than all repetitions of the sequence of instructions; and

providing an internal access point other than an externally called entry point and an external exit point, which internal access point isolates the one instruction within the sequence of instructions from only some of the repetitions so that the one instruction is within less than all of the repetitions.

18. The method of claim 17, wherein all of said steps are included within a step of storing a program.

19. The method of claim 17, wherein all of said steps are included within a step of transmitting a program.

20. The method of claim 17, wherein all of said steps are included within a step of receiving a program.

21. The method of claim 17, wherein all of said steps are included within a step of executing a program.

22. The method of claim 17, wherein all of said steps are included within a step of machine modifying a program.

23. A code rescheduler, comprising:

a storage media; and

means for rescheduling at least one instruction of a repeated sequence of instructions for execution by at least one and by less than all repetitions of the sequence of instructions.

24. A code rescheduler according to claim 23, wherein:

said means for rescheduling providing internal recursive access between an entry point and an exit point of the sequence of instructions.